

ЛЕКЦИИ МЕЖДУНАРОДНОЙ ШКОЛЫ «KAZCAS-16»

---

**Э.В. Чеботарева**

**Введение в конструирование и  
программирование электронных устройств  
на базе платформы Arduino**

---



---

Казань, Казанский федеральный университет, 5–7 ноября 2016

УДК 004.9+372.862

## ВВЕДЕНИЕ В КОНСТРУИРОВАНИЕ И ПРОГРАММИРОВАНИЕ ЭЛЕКТРОННЫХ УСТРОЙСТВ НА БАЗЕ ПЛАТФОРМЫ ARDUINO

Э.В. Чеботарева<sup>1</sup>

<sup>1</sup> chebotareva.elv@gmail.com; Казанский федеральный университет

*Лекция посвящена знакомству с платформой Arduino, с помощью которой можно сделать первые шаги в робототехнике. Приведено техническое описание и рассмотрены основы программирования платформы. Также в лекции разобраны примеры создания простых проектов.*

**Ключевые слова:** Робототехника, Arduino

### Введение

В настоящее время образовательная робототехника является актуальным и очень популярным направлением во всем мире.

Трудно переоценить те возможности, которые предоставляет включение в учебный процесс конструирования и программирования хотя бы одного физического устройства. Робототехника позволяет применить на практике знания, полученные учащимися на уроках физики, информатики, математики, симулирует развитие когнитивных способностей, мотивирует к творчеству и исследовательской деятельности, способствует приобретению навыков работы в коллективе. Кроме того, внедрение робототехники в учебный процесс стимулирует учителей к поиску новых приемов, способов и техник в обучении.

Настоящая лекция посвящена основам работы с конструктором Arduino. Arduino - это платформа с открытым исходным кодом, основанная на простых в использовании аппаратных средствах и удобной среде программирования. К достоинствам платформы также можно отнести сравнительно низкую стоимость и расширяемость как аппаратного так и программного обеспечения.

### 1. Аппаратные средства

Главной частью Arduino является печатная плата, с размещенным на ней микроконтроллером. В семейство Arduino входит несколько разновидностей плат [1], выбор той или иной платы обусловлен особенностями проекта. Например, простые проекты для начинающих рекомендуется выполнять с помощью платы Arduino UNO, с помощью платы Arduino MEGA можно создать сложные проекты, такие как 3D-принтер, а плата LilyPad Arduino предназначена для использования с предметами одежды и текстиля, с помощью нее, к примеру, можно разрабатывать «умную одежду».

Плата Arduino UNO основана на микроконтроллере ATmega328P. На плате имеется 14 контактов, которые могут быть использованы для цифрового ввода и вывода (порты 0-14 на рис. 1), 6 из них могут использоваться для аналогового выхода (порты 3,5,6,9,10,11 на рис. 1), 6 портов аналогового входа (A0-A5 на рис. 1). Платформа

Arduino UNO оснащена 32Кб флеш-памяти, оперативная память (SRAM) составляет 2Кб, также имеется 1Кб для долговременного хранения данных (EEPROM).

Питание осуществляется с помощью USB подключения или батарейки, при подключении через USB питание составляет 5 В. Диапазон входного напряжения составляет 7-12 В, процессор работает на напряжении 5 В. На плате имеются следующие контакты для доступа к питанию (рис. 1):

- Vin дает то же напряжение, что используется для питания.
- 5V дает напряжение 5В вне зависимости от входного напряжения.
- 3.3V дает напряжение 3,3В.
- GND - земля.

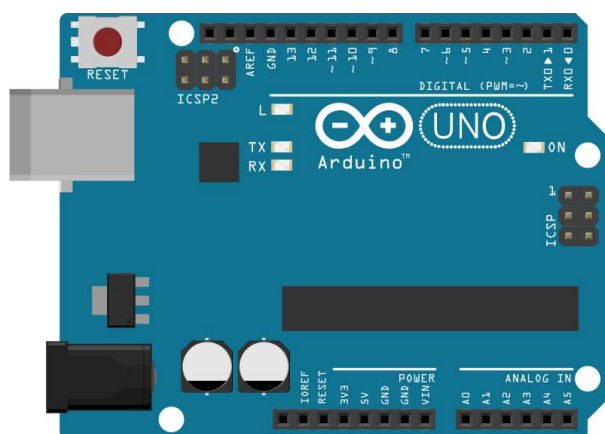


Рис. 1. Платформа Arduino UNO

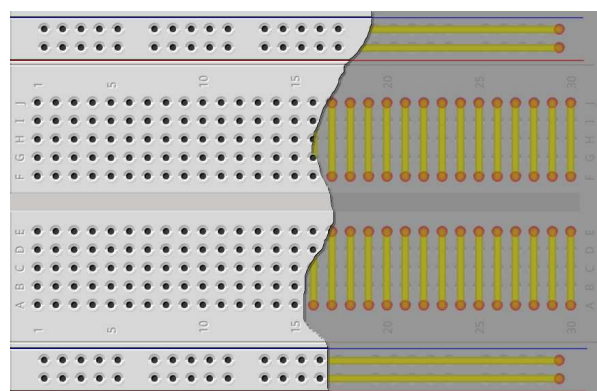


Рис. 2. Платформа Arduino UNO

В большинстве проектов используется так называемая макетная плата, которая предназначена для быстрой сборки схем (рис. 2). При использовании макетной платы не нужен паяльник, схему легко корректировать благодаря имеющимся на плате контактам, которые можно соединять с помощью проводов или перемычек.

К платформе Arduino UNO может быть подключено множество различных деталей и устройств. К ним относятся устройства вывода, управления, а также многочисленные датчики и сенсоры. На рис. 3 представлены те из их, которые понадобятся для реализации рассмотренных ниже проектов.

1. Резисторы. Резистор - пассивный элемент электрической цепи, обладающий сопротивлением.

2. Светодиоды. Светодиод светится при пропускании через него электрического тока в прямом направлении от анода к катоду. У светодиода две ножки. Аноду (+) соответствует длинная ножка светодиода, катоду (–) - короткая. Собственное сопротивление светодиода после насыщения очень мало, и без резистора, ограничивающего ток через светодиод, он перегорит.

3. Пьезоизлучатель. Пьезоизлучатель - это устройство, способное воспроизводить звук благодаря обратному пьезоэлектрическому эффекту.

4. Сервопривод. Сервопривод - это мотор, углом поворота вала которого можно управлять.

5. Фоторезистор. Фоторезистор меняет сопротивление в зависимости от количества света падающего на него. В полной темноте он имеет максимальное сопротивление, а по мере роста освещённости сопротивление уменьшается.

6. Ультразвуковой дальномер. Ультразвуковой дальномер излучает звуковой сигнал и принимает его отраженным от препятствия. Зная время, которое понадобилось звуковому сигналу чтобы преодолеть расстояние до объекта и вернуться после отражения от объекта, а также скорость звука, можно вычислить расстояние до объекта.

Также нам понадобятся соединительные провода, кабель USB, блок питания. Комплект деталей, на основе которого можно провести первые уроки по программированию Arduino приведен на рис. 3.

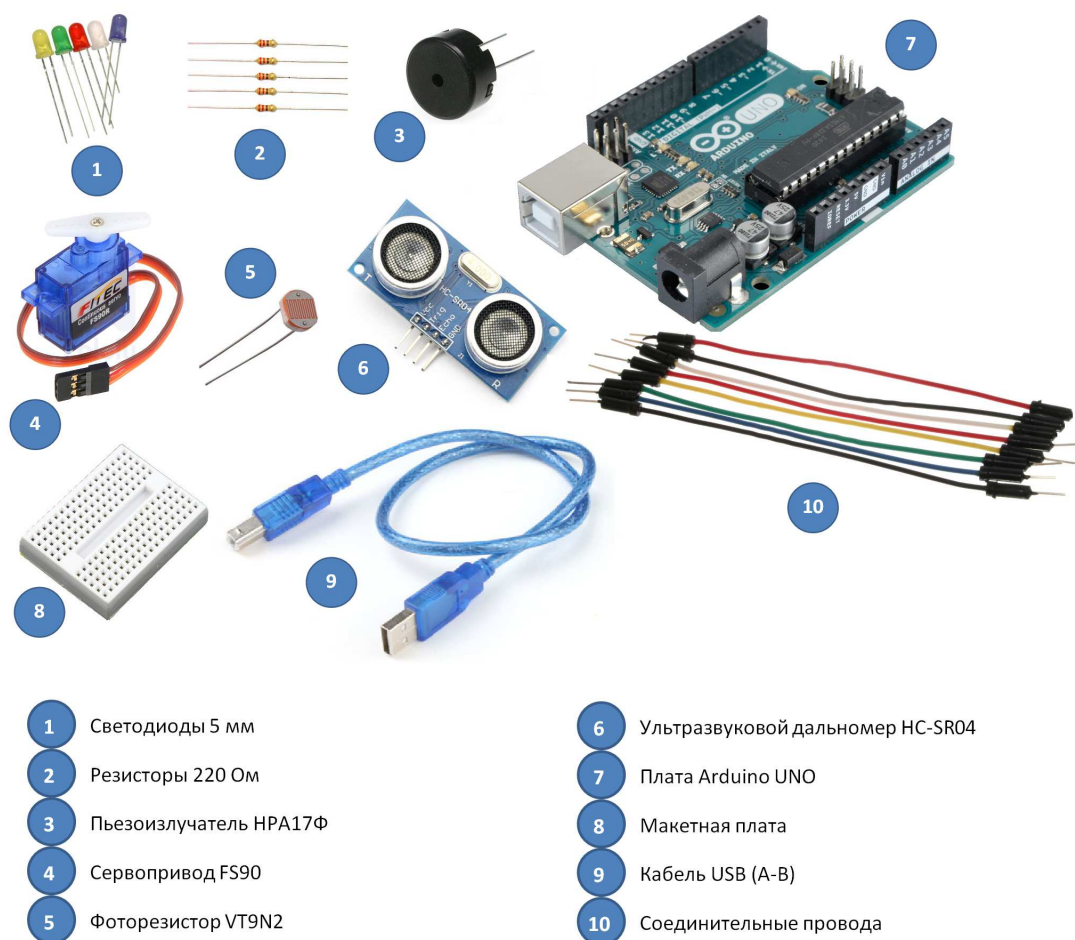


Рис. 3. Используемые детали

## 2. Программирование

Для программирования Arduino используется интегрированная среда разработки - Arduino IDE, которую можно скачать на официальном сайте проекта [2]. Языком программирования является C++. Программы для Arduino принято называть скетчами. На рис. 4 приведен вид окна Arduino IDE. Для компиляции скетча следует нажать кнопку «Проверить» на панели быстрого доступа, для загрузки скетча на плату

необходимо соединить плату с компьютером с помощью USB кабеля и нажать кнопку «Загрузить».

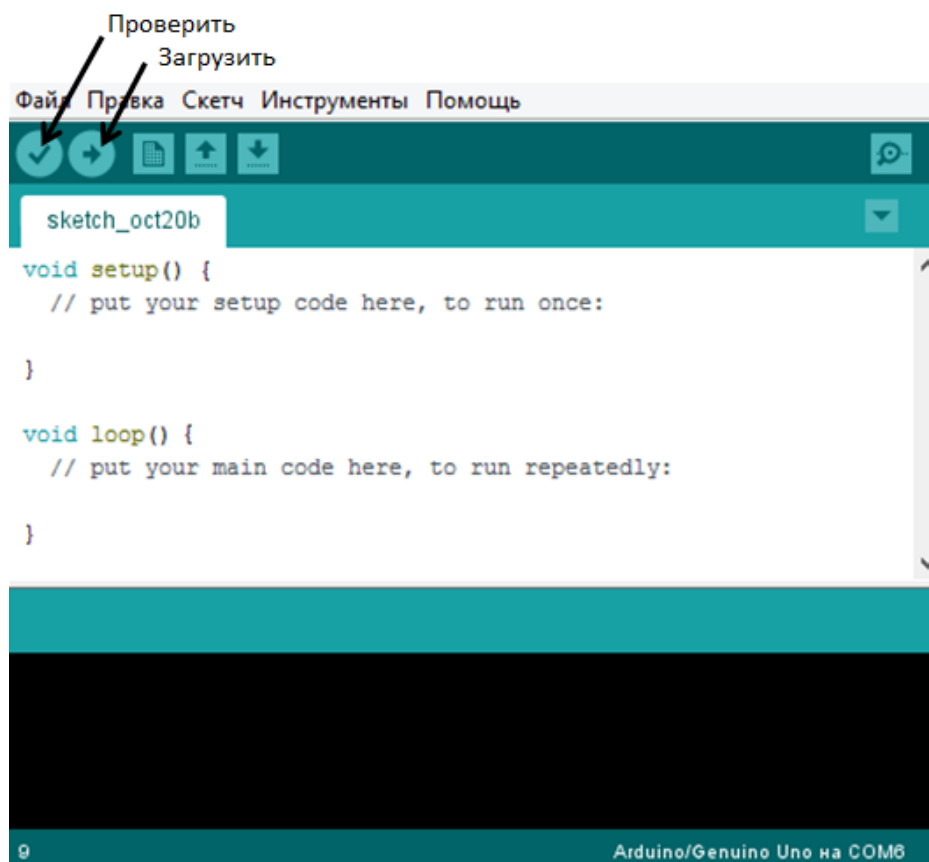


Рис. 4. Arduino IDE

Программа должна содержать две обязательные функции `setup()` и `loop()` (рис. 4). Функция `setup()` вызывается однократно при старте, а функция `loop()` выполняется в режиме бесконечного цикла.

### 3. Примеры простых проектов

#### 3.1. Мигающий светодиод

Первым проектом, который рекомендуется выполнить начинающим работу с Arduino является управление светодиодом.

Схема подключения приведена на рис. 5.

Для программирования проекта нам понадобятся функции, приведенные в табл. 1.1.

Код программы, для включения светодиода выглядит следующим образом.

```
void setup()
{
  pinMode(4, OUTPUT);
}
void loop()
```



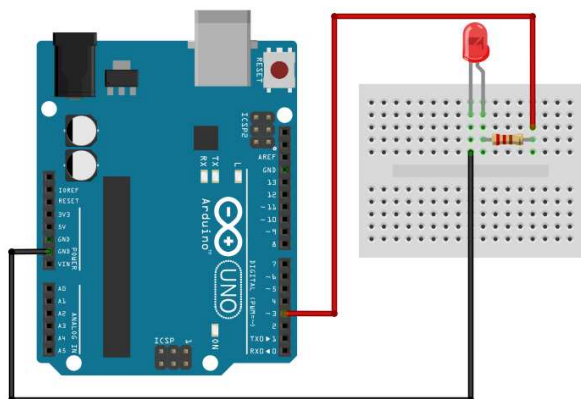


Рис. 5. Схема подключения светодиода

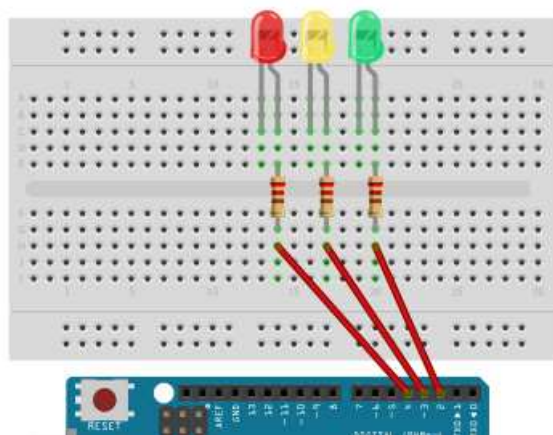


Рис. 6. Схема к заданию для самостоятельного выполнения

Таблица 1.1. Команды для проекта «Мигающий светодиод»

Функция	Краткое описание
<code>pinMode(pin, Mode)</code>	Устанавливает режим работы заданного цифрового вывода как входа или как выхода. <code>pin</code> - номер цифрового вывода, <code>Mode</code> принимает значения INPUT (вход) или OUTPUT (выход).
<code>digitalWrite(pin, value)</code>	Отправляет на цифровой вывод с номером <code>pin</code> значение HIGH или LOW.
<code>delay(ms)</code>	Останавливает выполнение программы на <code>ms</code> миллисекунд.

```
{
digitalWrite(4, HIGH);
}
```

Данная программа состоит из двух обязательных функций. Функция `setup()` выполнится один раз при запуске проекта. При этом функция `pinMode` установит цифровой вывод 4 в качестве выходного.

Функция `loop()` в режиме бесконечного цикла будет вызывать команду `digitalWrite`, которая подает на цифровой вывод с номером 4 значение HIGH, в результате чего светодиод включается.

Чтобы светодиод мигал, необходимо после включения обеспечить его выключение, подав на цифровой вывод с номером 4 значение LOW, а с помощью команды `delay` можно управлять временем ожидания между включением и выключением.

Код программы при этом примет следующий вид.

```
void setup()
{
pinMode(4, OUTPUT);
```

Таблица 1.2. Команды для проекта «Пьезоэлемент»

Функция	Краткое описание
tone(pin, frequency, ms)	Генерирует на выводе pin сигнал с частотой frequency и длительностью ms.
noTone(pin)	Останавливает сигнал на выводе pin.

```

}
void loop()
{
digitalWrite(4, HIGH);
delay(500);
digitalWrite(4, LOW);
delay(500);
}

```

Приведем также пример задания для самостоятельного выполнения. Требуется дополнить схему на рис. 6 и написать программу, которая включает и выключает светодиоды по очереди слева на право.

### 3.2. Пьезоэлемент

По аналогии со светодиодом можно управлять звуком, который издает пьезоэлемент. Для программирования проекта нам понадобятся дополнительные функции, приведенные в табл. 1.2.

Схема подключения приведена на рис. 7.

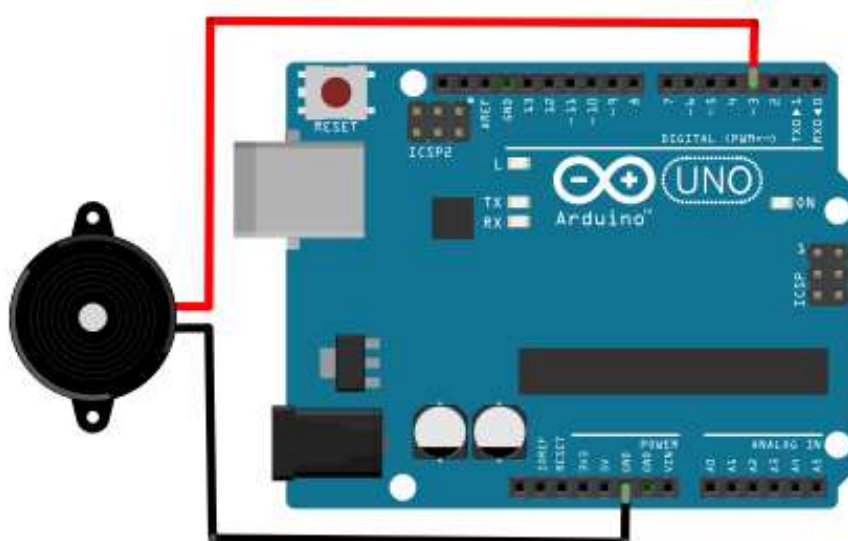


Рис. 7. Схема подключения пьезоэлемента

Чтобы получить два сигнала разной частоты, выполним программу.

```
void setup()
{
  pinMode(3, OUTPUT); //объявляем вывод 3 как выход
}
void loop()
{
  tone (3, 500,100); //сигнал частотой 500 Гц 100 мс
  tone(3, 1000,100); //включаем на 1000 Гц 100 мс
}
```

С помощью команды `tone` можно проигрывать различные мелодии, при этом `tone` лучше вызывать в массиве. В примере, приведенном ниже, проигрывается мелодия «К Элизе» Л. Бетховена.

```
const byte n = 35; // Количество нот
//массив частот нот
int notes[n] = { //массив частот нот
695, 622, 659, 622, 659, 987, 587,523, 440, 261,
329, 440, 493, 329, 415, 493, 523, 329, 659,622,
659, 622, 659, 987, 587, 523, 440, 261, 329, 440,
493, 349, 523,493,440};
int duration[n] = {//массив длительности нот
250, 250, 250, 250, 250, 250, 250, 250, 500, 250,
250, 250, 500, 250, 250, 250, 500, 250, 250,
250, 250, 250, 250, 250, 250, 500, 250, 250, 250,
500, 250, 250, 250, 1000};
void setup()
{
  pinMode(3, OUTPUT); //объявляем пин как выход
}
void loop()
{
  for (int i=0; i<=n; i++) {
    tone (p, notes[i], duration[i]); //проигрывание ноты
    noTone(p); //останавливаем звук
  }
}
```

Заданием для самостоятельной работы может служить реализация «цветомузыки», где в зависимости от частоты сигнала включается светодиод определенного цвета, а длительность сигнала совпадает с длительностью горения светодиода.

### 3.3. Сервопривод

Сервопривод FS90 позволяет установить и удерживать угол поворота колеса. Для работы с сервоприводом, его нужно подключить к плате согласно рис. 8.

Управлять сервоприводом можно с помощью библиотеки `Servo.h` (табл. 1.3). Для этого требуется создать переменную типа `Servo`, которая соответствует физическому сервоприводу.



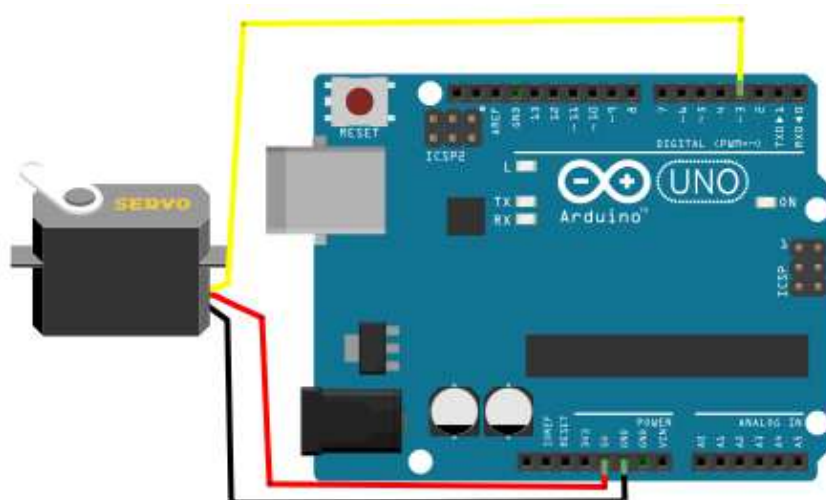


Рис. 8. Схема подключения сервопривода

Таблица 1.3. Команды для проекта «Сервопривод»

Функция	Краткое описание
<code>servo.attach(pin)</code>	Привязывает сервопривод с именем <code>servo</code> к выводу <code>pin</code> .
<code>servo.write(angle)</code>	Поворачивает вал сервопривода <code>servo</code> на угол <code>angle</code> в градусах.

```
#include <Servo.h>
Servo myservo;
void setup()
{
  myservo.attach(3);
}
void loop()
{
  myservo.write(0);
  delay(1000);
  myservo.write(180);
  delay(1000);
  myservo.write(90);
  delay(1000);
}
```

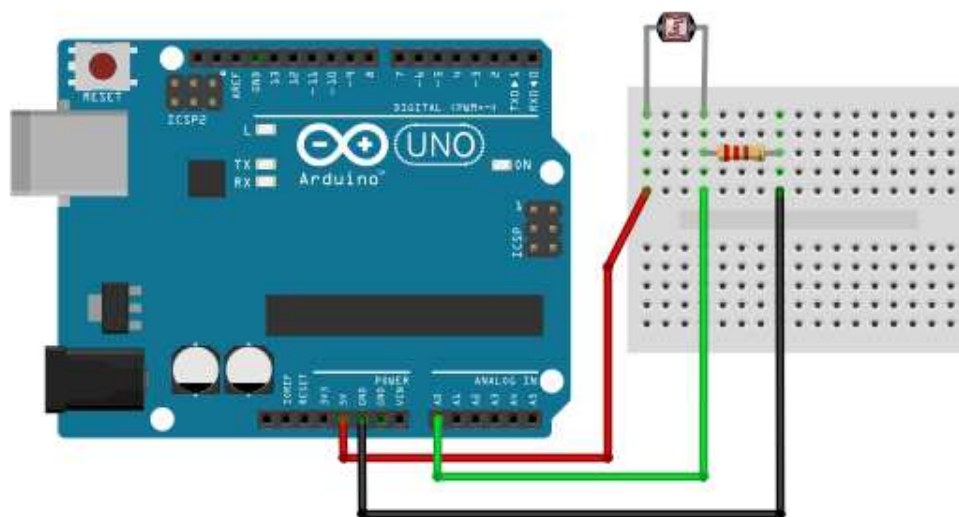
### 3.4. Измерение уровня освещенности

Фоторезистор - это резистор, меняющий сопротивление в зависимости от количества света падающего на него. Поэтому его можно использовать для измерения относительного уровня освещенности.

**Таблица 1.4.** Команды для проекта «Измерение уровня освещенности»

Функция	Краткое описание
<code>Serial.begin(speed)</code>	Открывает соединение и устанавливает скорость передачи равной speed бит/с.
<code>Serial.end()</code>	Закрывает последовательное соединение.
<code>Serial.print(value)</code>	Передает значение переменной value как текст на монитор
<code>Serial.println(value)</code>	Осуществляет при печати еще и перевод на новую строку.
<code>analogRead(Apin)</code>	Считывает значение с аналогового входа Apin.

Схема подключения приведена на рис. 9

**Рис. 9.** Схема подключения фоторезистора

Считываемые результаты можно выводить на монитор порта. Для организации связи Arduino с компьютером используют функции Serial (табл. 1.4) .

Код программы.

```
int illum;//уровень освещенности
void setup() {
  Serial.begin(9600);
  pinMode(A0, INPUT);
}
void loop() {
  illum=analogRead(A0)
  Serial.println();
  delay(20);}
```

Для просмотра получаемых данных требуется после загрузки проекта на платформу выбрать в окне среды разработки «Инструменты - Монитор Портов».

В качестве задания для самостоятельного выполнения предлагается собрать схему к следующей программе.

```
int illum;  
void setup() {  
  pinMode(A0, INPUT);  
  pinMode(5, OUTPUT);  
}  
void loop() {  
  illum = analogRead(A0);  
  if (illum<5) digitalWrite(7, HIGH);  
  else digitalWrite(7, LOW);  
}
```

### 3.5. Измерение расстояния

Ультразвуковой дальномер HC-SR04 имеет четыре вывода VCC, Trig, Echo, GND. Вывод VCC подключается к контакту 5V на плате. Вывод GND соединяется с контактом GND на плате. Через вывод Trig излучается звуковой сигнал, а вывод Echo принимает отраженный от препятствия сигнал. Выводы Trig и Echo можно подключать к любым пинам.

Пусть  $t$  - время (мс), за которое понадобилось звуковому сигналу чтобы дойти до препятствия и вернуться отразившись от него. Тогда расстояние  $l$  (см) до объекта от источника сигнала можно вычислить по формуле:

$$l = \frac{t}{58},$$

Функция `pulseIn(pin,value)` возвращает длину сигнала на выводе `pin` в миллисекундах, здесь `value` - тип ожидаемого сигнала, может быть HIGH или LOW.

Следующая программа выводит на монитор расстояние до препятствия, от которого был отражен сигнал. Выводы Trig и Echo подключены к пинам 4 и 3.

```
int echoPin = 3;  
int trigPin = 4;  
void setup() {  
  Serial.begin(9600);  
  pinMode(trigPin, OUTPUT);  
  pinMode(echoPin, INPUT);  
}  
void loop() {  
  int t,l;  
  digitalWrite(trigPin, LOW);  
  delay(2000);  
  digitalWrite(trigPin, HIGH);  
  delay(10000);
```

```
digitalWrite(trigPin, LOW);  
t = pulseIn(echoPin, HIGH); //время за которое звук вернулся  
l = t/58; //время в см. с учетом скорости звука  
Serial.print(cm);  
Serial.println(" cm");  
delay(100);  
}
```

В качестве задания для самостоятельной работы предлагается соединить дальномер с сервоприводом так, чтобы при обнаружении препятствия сервопривод поворачивал вал.

### **Литература**

1. Arduino Products: [Электронный ресурс] / Arduino, 2016. - Режим доступа: <https://www.arduino.cc/en/Main/Products/>
2. Download the Arduino Software: [Электронный ресурс] / Arduino, 2016. - Режим доступа: <https://www.arduino.cc/en/Main/Software>
3. Вики Амперка: [Электронный ресурс] / Амперка, 2010-2016. - Режим доступа: <http://wiki.amperka.ru>
4. Курс «Arduino для начинающих»: Занимательная робототехника, 2014-2016. - Режим доступа: <http://edurobots.ru/kurs-arduino-dlya-nachinayushhix/>